

## **Pinnacle V5.4 to V6 Direct Conversion Experience at UCSB**

Over the week of December 26<sup>th</sup> 2010-January 2<sup>nd</sup> 2011 we performed the “direct” conversion of our production V5.4 system to 6.4.1, so I thought I would share some of our experiences and my impressions. There is no denying that the upgrade process is a bit complex, it is also a fairly impressive technical achievement, as is also, I would say, Version 6.4. Version 6 brings what is essentially an “open architecture” (not the same as “open source”), which is really beyond a simple Application Programming Interface (API). It also delivers the availability of a very rich set of user-exits to allow custom business rule enforcement and the potential for a new level of automation. I believe these features put it ahead of the average software product of this type and size.

So, as I would expect would be the case with most sites considering the direct upgrade, there can be a lot that gets rolled in to this project. In our case it involved:

1. Running the “Pre-Verify” checks to clean up data integrity issues.
2. Upgrading from 5.4.4 to 5.4.11
3. Upgrading Oracle 10.4 to 11.1
4. Installing Oracle Apex on the Database.
5. Performing the 5.4.11 to V6.4.0 “direct” upgrade
6. Applying V6.4.1 maintenance (and very soon V6.4.2)
7. Set up and installation of a new Oracle Application Server (Oracle’s bundled Apache)
8. Installing Crystal Reports and Reporting Services engine on an Application Server (we did this on the same server as the Oracle App Server)
9. Updating local applications, re-fit local modifications (in to user exits), update reports, etc. This is actually what we spent most of our time on.

And while you’re at it, you might decide to move to a new piece of hardware, (V6 DOES take more resources) and upgrade your operating system, especially if you might even want to keep your existing system available as a contingency, if you’re lucky enough to have the hardware. In our case, we had the benefit of having an identical physical server which was fairly new, with Oracle licensing, on which to perform the DB upgrade. And I had the availability of a few healthy sized virtual servers. At the same time I had hoped to move to Windows Server 2008 R2 for both the DB server and the app servers (more on this later). My first piece of advice (which might be obvious):

- 1) It’ll be a lot easier if you have some servers (real or virtual) around to perform the testing, and building the new environment. [This probably doesn’t apply if you are in the Paetec hosted environment.]

## The Pre-Verify Process:

The upgrade process itself requires a bit of work cleaning up data prior to performing the upgrade. This is known as the “pre-verify.” This process runs over 300 (really!) checks on your data and identifies discrepancies in the V5 data. Trust me when I say that there is probably no one who will not have discrepancies. These issues must be corrected in your V5 data before undertaking the upgrade. Some of these are straightforward “one-time” fixes in your V5 data. And some of the issues identified will continue to pop up as long as you’re using the V5 system. We find that for whatever reason, we get a percentage of workorders that just plain do “funny” things with the data (personally I attribute some of these things to Omnis just not being quite up to the complexity of some of the things that people get in to, which is one of the many reasons moving to V6 is an important goal). So while many of the issues identified can be fixed in a straightforward manner, and possibly just in the client, you will, in all likelihood, need to do a bit of analysis on certain errors, and how to resolve them. You (or someone on your behalf) will need to examine the data using SQL queries, and you may end up resorting to some mucking about with SQL (in consultation with Paetec) to clean things up. On our first round of pre-verify, we were working with a clone of our production database. And so my next few pieces of advice:

- 2) For the pre-verify cleanup, work with a cloned copy of your database first, if you can. After you’ve performed the fix, and seen the impact, then put the fix in to your production database.
- 3) Start early on the pre-verify. I would suggest anyone considering the upgrade start as soon as possible so they can nibble away at some of the issues, and also to begin to understand why some of the issues arise. There may be something you need to consider doing differently with the product.
- 4) The pre-verify process will identify non-standard triggers. This a good start for finding custom code.
- 5) The pre-verify process identifies non-standard tables placed in the product schemas (AXIS, TELECOM, WORKORDER, CABLEMAN). These need to be moved, so this too can be done in advance of any conversion.

The diagnostics each have a diagnostic number (for which you can look up explanations on the Pinnacle website), and look something like:

```
1814-PREVERIFY_640_1814 - DATA INTEGRITY: In ASSN_ITEM, not in
SERVICE_CHARS - OK
1815-PREVERIFY_640_1815 - DATA INTEGRITY: In SERVICE_CHARS, not in
SERVICE_LOCATION - 1 record(s) found
    0/18245/A/D: ,M/S: , SA: //
1816-PREVERIFY_640_1816 - DATA INTEGRITY: In SERVICE_LOCATION, not in
SERVICE_CHARS - OK
```

- 6) For each of these issues, you will need to perform some activity. So, I would suggest coming up with a scheme for organizing your procedures (screen shots) and SQL scripts you used to deal with each issue. You may be doing it again (and possibly again).

### **Getting to V5.4.11 and Oracle 11.1 and Apex**

So next, one must get to V5.4.11, the latest V5 version, and one that runs both on Oracle 10.2 and 11.1. Prior to V5.4.11, Pinnacle was not “Certified” on Oracle 11.1. And V6.4 must run on 11.1. So, one is put in the position of needing to upgrade to Pinnacle V5.4.11, then upgrade to Oracle 11.1, then perform the V6 upgrade. If you routinely upgrade your Pinnacle versions, and moving to 5.4.11 is no big deal, and then you can leisurely upgrade to Oracle 11.1 you should certainly take that route. Because we had a fair bit of customization (and some fairly complex ones in the portal) in our V5.4.4, we had chosen several years ago in anticipation of V6 was to freeze it at that level. So, the path we needed to take was to perform the V5.4.4-V5.4.11 upgrade as part of the upgrade itself. Fortunately, the V5.4.11 upgrade seems fairly well behaved. I wish I could say the same of Oracle upgrades. I’m sure that those of you with the benefit of “industrial strength” DBAs, because you run it for other applications, aren’t daunted by Oracle version upgrades, but I personally, find it to be a messy business. In the past, moving between versions, (we started with Oracle 7), I have always brought up a clean new DB (usually at the same time I’ve upgraded hardware, and so had a new box on which to build the system), and then performed (logical) exports and imports. (There is, however, always a little bit of cleanup (grants etc.), required moving the DB around.)

On the first trial upgrade (you will probably do 2 or so), I made a clone of my Oracle 10 system on to my standby box, performed the V5.4.11 upgrade, performed pre-verify cleanups, and then performed the Oracle 11.1 in-place upgrade. On my second trial upgrade and production upgrades, I resorted to the path which I personally find more reliable, and instead performed a full-import of my V5 DB in to a fresh Oracle 11.1 instance. However, this put me in the situation of actually doing my V5.4.4 (remember it’s not officially supported under Oracle 11.1) to V5.4.11 under Oracle 11.1. After some arm twisting, it was agreed that after importing V5.4.4 to Oracle 11.1, if V5.4.4 integrity checks and the V5.4.11 upgrade were both clean, we could use this strategy. So my next few pieces of advice:

- 7) If you are moving from Oracle 10 to Oracle 11 as part of this process (that is, if you are not already on V5.4.11 and Oracle 11.1), do not underestimate the issues regarding Oracle upgrades. While my in-place Oracle 10 to 11.1 upgrade was adequate for testing, I personally would not have wanted to use it for production (your experience may vary).
- 8) Fully understand the prerequisites for the different versions of the Pinnacle product, the Oracle versions (and APEX), and the OS. Oracle 11.1 is not the latest version of Oracle (V6.4.2 does support Oracle 11.2 I believe). The latest version of the OS (In my case I was hoping to use Windows 2008R2) might not be supported by 11.1. This applies to bringing up the App Server as well (more later.)
- 9) Make sure you have cleaned up any existing APEX installation(s) in the DB prior to installing the required version of Apex. (I had to back up several steps on my first trial to clean this up).

- 10) Be on very good terms with your DBA, or whoever is going to do all the Oracle backend work, the project requires a fair bit of work from the DBA.
- 11) [A suggestion to Paetec: While obviously there are a lot of different configurations out there, some of us out here only use Oracle for this application. So a little more guidance on specific versions and packagings of Oracle software would be really helpful).

### **The Direct Upgrade process itself:**

The Upgrade process itself involves running some 15 steps, each of which run a lot of processing. Your RATED file, in particular will require a lot of processing. The logs from each step are sent to the Pinnacle upgrade team, and the specialist assigned to your upgrade. Despite all of the pre-verify activity, there may be still be errors reported, it is after all, a fairly complex application, with a lot of data dependencies. One of the issues in terms of how quickly you can move through the upgrade is turnaround on these steps. Perhaps some sites might just have Paetec staff remotely running the steps and analyzing the results [I don't know if that is an option], but I think if you are performing the upgrade steps yourself, you need to take in to account that is probably not realistic to expect staff at both ends to simply be sitting at their keyboards waiting for the emailed logs or corrections to arrive. So there is some latency in the processes. Though I believe Paetec is also working on even further automating the process.

- 12) During the first conversion test, you might find errors sufficient to require restoring to a previous step. Be prepared for it.
- 13) Reduce the size of your RATED file as much as practical. If you need the records around for analysis, consider if you really need them to remain available in the product, or if you can just copy some of it out, but still keep it online. You should check with Paetec first to be sure you don't lose important history. (BTW: According to Paetec, V6 does now support an archive process for some of the most sizeable data (specifically, the rated (billable usage), billing, and accounting information.)
- 14) If you are running in Oracle Archive Log Mode (you are aren't you?), you should seriously consider turning this off during the upgrade process. Not only does it slow things down, but it can generate a tremendous amount of transaction logging, perhaps using more disk space than you might have available (and when you run out, it all stops). You should instead make backups between steps (or at least the major steps), to allow you to roll back to a previous step if there is a problem with a step, and turn off archiving.
- 15) To expedite resolving issues, it is beneficial for Paetec to have access in to your database. If you have a "robust" firewall administration process (bureaucracy), you should get that straightened out early.

### **Setting up the Application Server and Reports Server**

V6, as you know, is all web (browser) based, there is no “back-office” client application. Instead, you set up a special application (web) server, which interacts with the database. I believe there are some sites out there that have never actually used the prior version’s Web Portal to Pinnacle. So, getting an application server set up may be a new step for some sites. In addition, the reporting services (Crystal Reports engine) provides a significant bit of functionality within the product, providing the majority of reports, and the “dashboard” (graphical) reports. So while it is still “optional”, it is becoming increasingly valuable. In V5 and prior, some sites, even those with the web portal, may not have had the Crystal Reports component installed. This can be installed on the same server as the web server (if that is on Windows), or on a separate server.

- 16) There are a number of different versions of the Oracle packaging of the Application (web) Server. (More than of the database itself). As with the Oracle Database, you must pay particular attention to your combination of versions and platforms, and will probably want to seek confirmation of the combination you’ve selected. I discovered, painstakingly, that Crystal reports works only with the 32 bit of Windows Oracle, not the 64 bit version. You will also need a Java Runtime component, and a Pinnacle provided Report Server Component (which services requests from the database).
- 17) Because you are exposing your application server to the “world” (or at least your campus), you may well have the application server on a separate network from your database. And if so, you will probably have a firewall (or more) in between. Obviously, the application server and the report server will need access through the firewall to the database (see item 15). Not so obviously, the database server will initiate conversations with the Pinnacle report server engine. Ensure you have the firewall(s) configured accordingly. (Tip: one way to verify that you can access the report server engine from the database machine is to “telnet” from the database server to the report server specifying the port at which you have configured the Pinnacle report server to service. E.g. “telnet reportserver 8080”). Also, the report server keeps logs down in its windows “Program Files [(x32)]” directory.

### **Getting acquainted with V6.**

- 18) Learning the new navigation can be a bit of a challenge to users accustomed to the old system. It is important in training to reinforce the hierarchical structure of the “Menus”, “Tabs”, “Tab Items”. The online help within the application has several videos that help reinforce these important concepts.
- 19) Your V5 security templates are not brought forward. There is a new concept of Roles (which are, actually, easier to configure). If you plan to make any changes to the provided security Roles you must make your own local copies.
- 20) Even if you are not concerned about the “security” of having all of the buttons and tabs available, you should consider removing the Menus, tabs, tab items for unused areas of the

system, such as “service items”, “service features”, “interconnects” etc. simply to reduce the potentially overwhelming number of options available to the user. For configuring this, you can manipulate the available tabs, and also configure the “Privileges” associated with a role. I have noticed however, that in disabling certain tabs etc. there can sometimes be unexpected side-effects, so it does seem necessary to look things over fairly well even if you are making a seemingly minor change.

- 21) For “Back-Office” users, I would suggest starting simply with the Roles. Perhaps start with a base local Role, and eliminate all of the functions you simply do not use. Then make copies for the functional areas, to further eliminate areas not used for those functional areas.
- 22) For sites already using the public web portal, customer access to departments and accounts is based on Filters. The conversion migrates the V5 Contact assignments (department, accounts) over to unique filters for each user. You may find a need clean these up later, and load “generic” department filters, to ease future maintenance.
- 23) The V6 documentation set can be a bit daunting. Navigating the documentation, and the coverage on some topics, are areas in need of some strengthening [Paetec has stated they are working on this]. The V6 documentation area on the Paetec web site does not provide the ability to search the documentation. Fortunately, you can download the whole set. You can then use your own search tools. We put them on our Sharepoint site to allow searching. I would certainly recommend doing something like that.
- 24) I don’t think it is realistic to expect your backoffice users to use the documentation set as provided. I don’t think Paetec expects that, many of the documents are set up to be customized. I ended up simply creating “from scratch” a training document that essentially followed through the major work processes (scenarios) with screen shots of the product and narrative. This was actually not as bad as it sounds, since you will most likely be running through all the processes to validate them.
- 25) There seem to be a lot of little setup items that didn’t make it in to the “Post-Upgrade Activities” document. Keep a good log (with screenshots) of the configuration changes you make in your test instances. It might seem obvious when you do it, it’s surprising how easy it is to lose track.

### **Converting custom applications and the APIs and Exits.**

One of the big strengths of V6 is the availability of APIs (Application Programming Interfaces) and User Exits. The API’s are essentially database “views” of the underlying data, with the programming logic necessary to make it all hang together “inside” those views as “triggers”. This provides the ability in many cases to allow outside applications to safely populate data structures within the product. This goes far beyond imports, because updates (and in some cases deletes) can be made by external code directly against the core data structures, and the inter-relationships are taken care of. Unlike the more traditional API of most vendor products, which often only expose a small part of the product or its entities, the Pinnacle API exposes virtually the entire product. In fact, the web application uses these

same API views to a large extent. There are however, some functions that still require procedures (not necessarily documented to customers) which certain activities utilize. The API views do not however, necessarily mimic the structure of the underlying tables. In particular, there was an extensive effort to clean up column naming to achieve more consistency. So, any custom code you may have developed does need to be worked over to use the new views with their new names. Further, while the views are documented with a description of the purpose, and column descriptions, there is not yet an abundance of information as to which fields may be required or may automatically be populated, so custom development, while a great step forward from previous versions, still requires some research and development work. Paetec has stated that they intend to continue enhancing this documentation.

The User Exits provide the ability to enforce business rules, perhaps editing data, or checking against other data, and can even allow additional records (in other structures) to be populated or updated based on changes to a record. For example, we have an OCC for each Commodity in our inventory. When we change the cost of a commodity, we want the OCC cost to be updated. Previously, we had implemented (at our own risk) our own database trigger to perform this action. With the Exit, there is a defined and documented method of performing this. Another benefit of the exits is that according to Paetec, custom exit code should be fairly immune to release upgrades. The exit can manipulate structures using the API views. If you are converting legacy custom trigger code, as noted above, the API views have had a lot of column names "overhauled." So, while much of the underlying logic of our triggers can be essentially "ported" there is a bit of work required to map the fields and field names. But, this is now a far safer endeavor than before, and the application much more gracefully handles the situation when you want to "reject" a change.

There are also the PINN\_REPORT\_API reporting views. These are mostly the same as in V5. However, in the interest of consistency it seems, a few fields have been renamed. So, while we have made efforts since V5 to use the PINN\_REPORT\_API views for reports, even those reports were not immune to needing changes.

One of the areas much changed in V6 is the "Rating" engine, which is now far more flexible and programmable, and has been termed the "Flexible Rating Engine". One of the impacts of this change is that the classic RATED table has undergone some significant changes as to its use. Particularly, at least with our new Rating routine, a number of the columns in RATED are no longer populated (the data are elsewhere). So, reports you might have that are reporting directly against RATED will be impacted.

Another of the significant changes "under the covers", which may affect some users that have custom reports is the elimination of the Oracle "Public Synonyms" for Pinnacle tables. Previously, it was not necessary to know the schema (table owner) of a particular table. If you wrote a query to interrogate the RATED table, it was not necessary to specify "TELECOM.RATED". Unfortunately, it is not always intuitive to which schema a table belongs. And many of us (at least here) have gotten lazy and let Public synonyms resolve these names. So, for those of you similarly disinclined to specificity, you may have a bit of cleanup work. Well, we have hundreds of reports and SQL queries. Some are not used all that often. We simply did not have time to visit nearly enough of them, so we took a rather inelegant approach and created "legacy synonyms" for the V5 tables. We created private synonyms in the user

schemas for those users and applications which run reports or queries. This is actually what Paetec has done in its own schemas. That is, in the WORKORDER, CABLEMAN, AXIS schemas there are synonyms for RATED pointing to TELECOM.RATED. The short term problem with this little short-cut (and it was already apparent with just Paetec's synonyms), is that by creating synonyms for so many tables in so many schemas, the handful of people who had rights to select from any table in the database found some of their database tools no longer happy finding tens of thousands of what appeared to be tables to choose from.

Our plan is to go back and eliminate these synonyms after some conversion period. Paetec has stated its intention to eliminate the "legacy synonyms" in their schemas in a future version. And while most of the base tables have remained largely intact, there are legacy columns that are no longer used, and some may be eliminated in the future as Paetec continues to clean things up and enhance performance. So clearly a part of the conversion is that sites with custom reports should be moving reports to the API views and away from the tables themselves. So, from all of this a few bits:

26) Today, whenever making new reports, utilize the PINN\_REPORT\_API views.

27) Convert all unqualified table names to fully qualified table names.

28) For custom reports and applications, consider using custom intermediate views to isolate the reports and applications from the structures (tables or views) below. For reporting, ideally you can have a fairly small set of general purpose reporting views. Those views could (should) still use the PINN\_REPORT\_API views, which will have some changes, but at least it is a lot fewer places to change.

29) If you have a lot of customizations and reports, unless you have copious amounts of staff and time to convert all of them prior to conversion, perform a triage of your customizations to determine which must be converted prior to conversion. It is also an opportunity to revisit or reengineer the processes that created the need for the customization.

## **Summary**

There is no question that this project can be a lot of work. But the Paetec conversion support staff have been excellent, knowledgeable, and easy to work with. While a conversion from V3 or V4 to V5 was a large project, the resulting environment was still very familiar. This change in user interface is not trivial for users who have grown very accustomed to the old product. However, I think most everyone here would agree that once you do get used to the new interface, it is much more consistent and accommodating. Being a web application allows it to be deployed with far more ease than the Omnis client for back-office users. This also offers the potential to simplify deploying back-office functionality across organizational lines. And the ability to configure Roles and Filters in combination with user exits, we expect, will allow us to provide far more functionality directly to our customer users. V6 is really a huge step forward, and despite the work, for our organization it has been well worth the effort of upgrading.

## **DISCLAIMER OF ENDORSEMENT AND LIABILITY**

**Disclaimer of Endorsement:** Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the University of California, and shall not be used for advertising or product endorsement purposes.

**Disclaimer of Liability:** Neither the University of California, nor any of its employees, makes any warranty, express or implied, including warranties of merchantability and fitness for a particular purpose, or assumes any legal liability for the accuracy, completeness, or usefulness of any information from this server or the links to servers on the site.